

INTEGRANTES

En equipos de 4 personas: Todos los integrantes deben estar en capacidad de poder sustentar completamente cualquiera de los aspectos del proyecto. El profesor se reserva el derecho de calificar grupal o individualmente.

PROYECTO FINAL

Elaborar un programa en PYTHON que resuelva laberintos con las siguientes características:

1. El laberinto debe ser cuadrado (igual número de filas y de columnas).
2. El programa deberá preguntar al usuario el tamaño del laberinto. Debe funcionar para cualquier tamaño de laberinto.
3. El programa deberá leer del usuario el laberinto como un String de caracteres concatenando las filas. El laberinto se expresará con las siguientes claves:
 - a. **O** : Posición inicial
 - b. **X** : Posición final
 - c. **+** : Paredes. (No transitables/obstáculo)
 - d. **" "** : (Espacio en blanco) celda transitable

Ejemplo: Un laberinto de tamaño 5: El String de 25 caracteres que servirá de entrada al programa para describir el laberinto sería algo como: "□□+□□+□□□+□□+□□□++□□X+O□", (donde □ es un espacio transitable)

		+		
+				+
		+		
	+	+	+	
	X	+	O	

4. El programa resolverá el laberinto de forma AUTOMÁTICA.
5. La opción Automática encontrará la solución del laberinto por medio de un programa que los resuelva mostrando el camino al dejar en cada posición de la ruta una letra "o" minúscula y la actual posición de la tortuga "@". Si se regresa para explorar una nueva ruta, debe despintar la "o" y dejar de nuevo el espacio en blanco.
6. Con el objeto de ver el funcionamiento del programa, deberá pintar el laberinto cada vez que avance o retroceda una posición en la búsqueda de la solución. Para lograr este efecto debe investigar cuál es la instrucción que detiene la ejecución durante un tiempo, antes de continuar su ejecución. (Antes de comenzar a resolver el laberinto, el usuario debe especificar el número de milisegundos a esperar entre paso y paso)
7. Para pintar el laberinto debe pintar el marco como se indica en el ejemplo a la derecha y entre cada posición horizontal del laberinto, debe pintar dos espacios.
8. El programa debe encontrar todas las posibles soluciones y pintar cada solución (mostrar el tablero del laberinto con la solución).
9. Debe utilizar las estructuras de datos que requiera entre las cuales puede usar colas, pilas, árboles, listas enlazadas, otras o combinaciones de ellas.
10. Después de encontrar y mostrar todas las soluciones, debe mostrar:
 - a. La solución más corta
 - b. La solución más larga
 - c. Tiempo promedio para encontrar una nueva solución
 - d. Tiempo para encontrar la solución más corta
 - e. Tiempo para encontrar la solución más larga
11. Informe del proyecto el cual debe incluir:
 - a. Portada
 - b. Objetivos del proyecto (Propios del equipo)
 - c. Descripción de la solución explicando las estructuras de datos utilizadas
 - d. Análisis de complejidad y orden con notación BigO de cada parte del programa.
 - e. Diccionario de Variables y métodos definidos en el programa
 - f. Documentar un ejemplo de laberinto de 10x10 y sus salidas
 - g. Conclusiones de acuerdo con los objetivos
 - h. Referencias (Todas las utilizadas incluyendo la AI, si fue utilizada)
12. El programa y el informe deben ser enviados al correo del profesor gabriel.soche@unisabana.edu.co a más tardar el martes 19 de noviembre.
13. La sustentación la harán los integrantes del grupo el miércoles 20 de noviembre en el horario que se programará con anticipación.

Total de movimientos : 19
+-----+
| + |
| + o o o o + |
| o o + o o |
| o + + + o |
| o @ + O o |
+-----+
Finalizó en 12 pasos

Rúbrica.

En esta asignatura se evaluarán las competencias de Diseño y Conducir Experimentos.

Item	Porcentaje
Informe	20%
Sustentación	60%
Programa Funcional:	20%
El programa funciona sin errores	2%
Presenta todas las soluciones en la forma solicitada	5%
Presenta los tiempos	5%
Presenta solución más corta y más larga	5%
Usa más de una estructura de datos	3%